



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

# Nhập môn Công nghệ phần mềm Introduction to Software Engineering (IT3180)

## **Bài tập tuần 08**

Thiết kế phần mềm  
(Thiết kế kiến trúc & thiết kế chi tiết)

Giang vien

# Mục tiêu

- Thực hiện các bài tập (câu hỏi) về các khái niệm và nguyên tắc thiết kế cơ bản trong Thiết kế phần mềm
- Hiểu được phân loại thiết kế kiến trúc, các khái niệm gắn kết (cohesion) và ghép nối (coupling)
- Xây dựng kiến trúc cho phần mềm của bài toán case study trong môn học:
  - Sử dụng package tổ chức các class trong mã nguồn phần mềm
  - Thiết kế cơ sở dữ liệu
  - Xây dựng các lớp model và các lớp truy xuất dữ liệu trong Database

# Đánh giá

- Hoàn thành các bài tập về các khái niệm và nguyên tắc thiết kế cơ bản trong Thiết kế phần mềm
- Hoàn thành import project do giảng viên cung cấp để tổ chức các class trong mã nguồn phần mềm
- Thiết kế CSDL trên hệ quản trị MySQL
- Hoàn thành xây dựng các lớp model và các lớp truy xuất dữ liệu trong Database

# Bài 1.1

**a) Kiểu kiến trúc ống dẫn – bộ lọc (pipes and filters) là một dạng đặc biệt của kiểu client-server?**

- 1. Sai**
- 2. Đúng**

# Bài 1.1

**b) Yếu tố nào có ảnh hưởng đến việc thiết kế kiến trúc phần mềm?**

**1.Công nghệ sử dụng**

**2.Yêu cầu về thuộc tính chất lượng**

**3.Chiến lược triển khai hệ thống**

**4.Tất cả các phương án trên**

**c) Chọn những yếu tố quan trọng tạo nên kiến trúc phần mềm (chọn nhiều)?**

- 1. Phần tử phần mềm**
- 2. Mối quan hệ giữa các phần tử**
- 3. Thuộc tính của mối quan hệ**
- 4. Thuộc tính của môi trường hoạt động**
- 5. Thuộc tính của phần tử**

# Bài 1.1

**d) Mẫu thiết kế chỉ áp dụng được cho lập trình hướng đối tượng?**

- 1. Đúng**
- 2. Sai**

# Bài 1.1

**e) Một số lớp có được trong giai đoạn phân tích có thể được lược bỏ trong giai đoạn thiết kế?**

- 1. Đúng**
- 2. Sai**



# Bài 1.1

**f) Trong thiết kế theo phương pháp cấu trúc, tương tác giữa các chương trình con (subprogram) được thể hiện trong ...?**

- 1. Biểu đồ cấu trúc**
- 2. Biểu đồ dòng dữ liệu vật lý (physical DFD)**
- 3. Biểu đồ dòng dữ liệu logic (logical DFD)**
- 4. Tất cả các phương án trên đều sai**

# Bài 1.2

- a) Trình bày ba loại biểu đồ thường được dùng trong phương pháp phân tích và thiết kế cấu trúc?
- b) Cho ba ví dụ về ba loại biểu đồ UML khác nhau thường được sử dụng trong phương pháp thiết kế hướng đối tượng?
- c) Trình bày mối quan hệ giữa các giai đoạn phân tích - thiết kế - thực hiện - kiểm thử - bảo trì. Hãy liệt kê các công việc cần được thực hiện ở giai đoạn thiết kế phần mềm. Vì sao nói “Trên thực tế phân tích và thiết kế không có sự tách biệt nhau mà hai giai đoạn này được tiến hành song song và bổ sung cho nhau”?

# Bài 1.2

a) Trình bày ba loại biểu đồ thường được dùng trong phương pháp phân tích và thiết kế cấu trúc?

## Gợi ý:

- Biểu đồ luồng dữ liệu (DFD)
- Biểu đồ thực thể - quan hệ (ERD)
- Biểu đồ cấu trúc (SC)
- ...

# Bài 1.2

b) Cho ba ví dụ về ba loại biểu đồ UML khác nhau thường được sử dụng trong phương pháp thiết kế hướng đối tượng?

## Gợi ý:

- Biểu đồ tuần tự
- Biểu đồ lớp
- Biểu đồ trạng thái
- ...

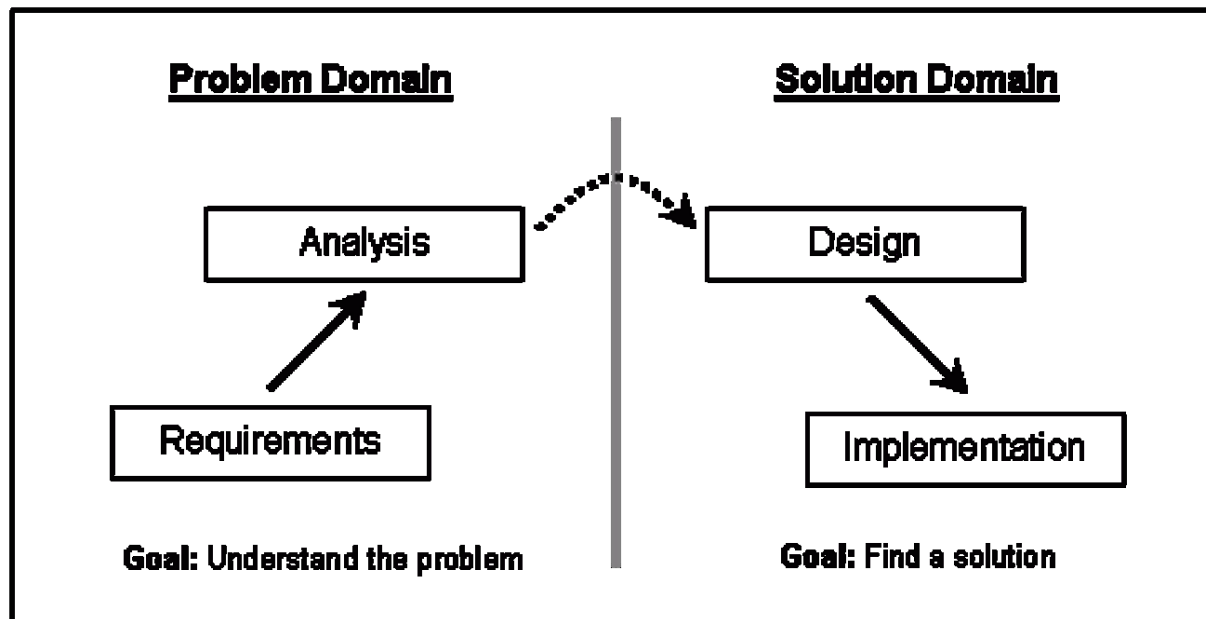
# Bài 1.2

c)...

Kết quả của quá trình thiết kế là  
**Software Design Document (SDD)**

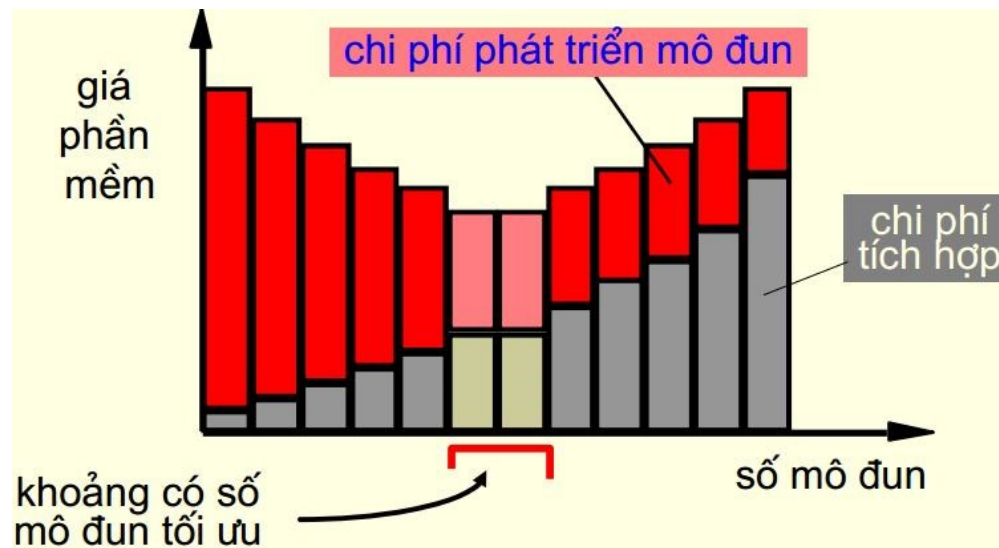
## Gợi ý:

- Phân tích nhằm trả lời câu hỏi “**cái gì**”, còn thiết kế để trả lời câu hỏi “**như thế nào**”.
- Hoạt động chính của pha thiết kế là tiến hóa tập biểu diễn phân tích thành tập biểu diễn thiết kế



# Bài 1.3

a) Trình bày khái niệm mô đun. Thông qua hình dưới, hãy phân tích và giải thích vì sao phải phân chia mô đun một cách tối ưu (số lượng mô đun nhiều quá hoặc ít quá thì không tốt, các yếu tố ảnh hưởng: giá PM, độ phức tạp, công sức thực hiện,..)?



# Bài 1.3

a)...

## Gợi ý:

- Mô đun hóa là một kỹ thuật chia hệ thống phần mềm thành nhiều mô đun rời rạc và độc lập nhau. Các mô đun sẽ có khả năng thực hiện công việc một cách độc lập.
- Việc phân chia mô đun thành những mô đun nhỏ hơn thì công sức để thực hiện các mô đun giảm xuống và như thế độ phức tạp cũng giảm theo nhưng công sức tích hợp các mô đun có chiều hướng tăng lên.
- Cần xác định số mô đun tối ưu trong quá trình phân chia (số lượng mô đun không quá nhiều, không quá ít)

# Bài 1.3

**b) Khi thiết kế phần mềm, việc che giấu thông tin và phân chia mô đun có quan hệ như thế nào? Vì sao trong thiết kế phần mềm cần thực hiện việc che giấu thông tin? Hãy liệt kê các lợi ích của việc che giấu thông tin.**



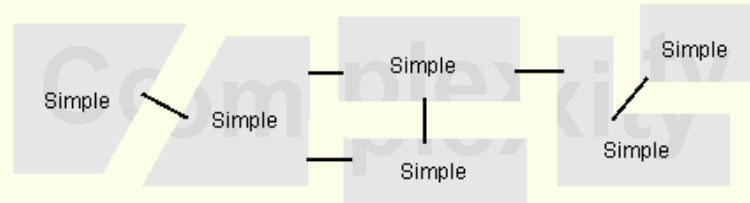
# Bài 1.3

b)...

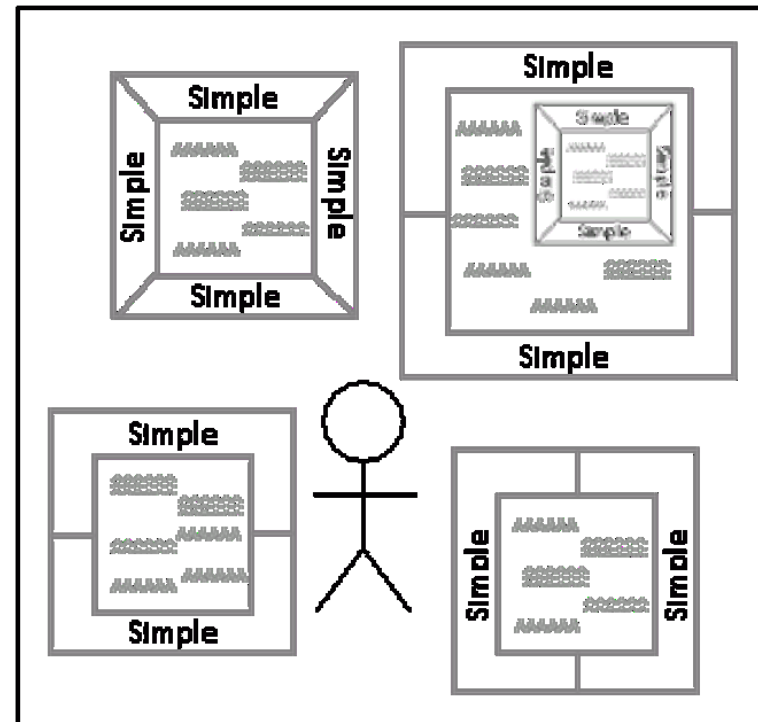
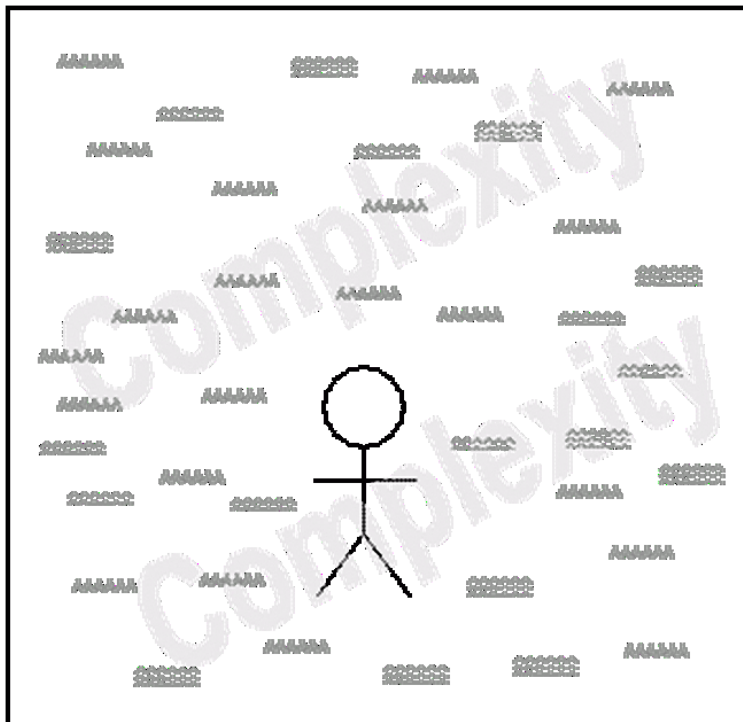
Gợi ý:

## Complexity

=>



- Một thiết kế tốt sẽ ẩn giấu sự phức tạp bên dưới các giao diện đơn giản



# Bài 1.3

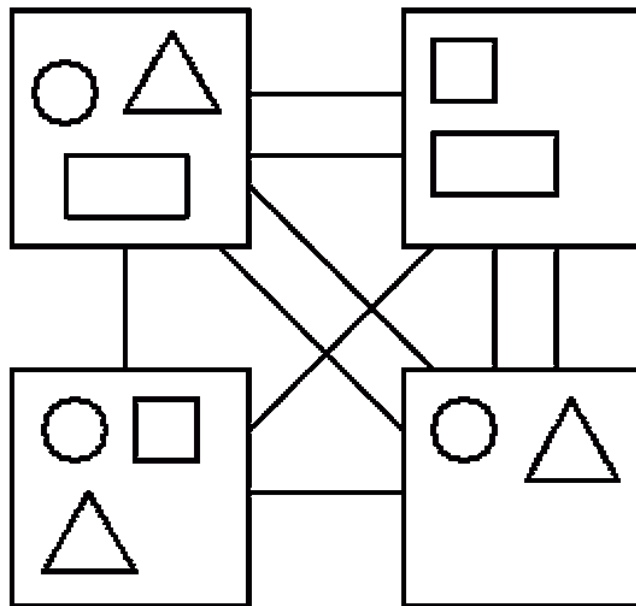
**c) Trình bày đặc điểm của tính ghép nối và kết dính trong phân chia mô đun. Yếu tố ghép nối và kết dính giữa các mô đun có ý nghĩa gì trong thiết kế? Vì sao khi phân chia mô đun phải tăng độ kết dính và giảm sự liên kết?**

# Bài 1.3

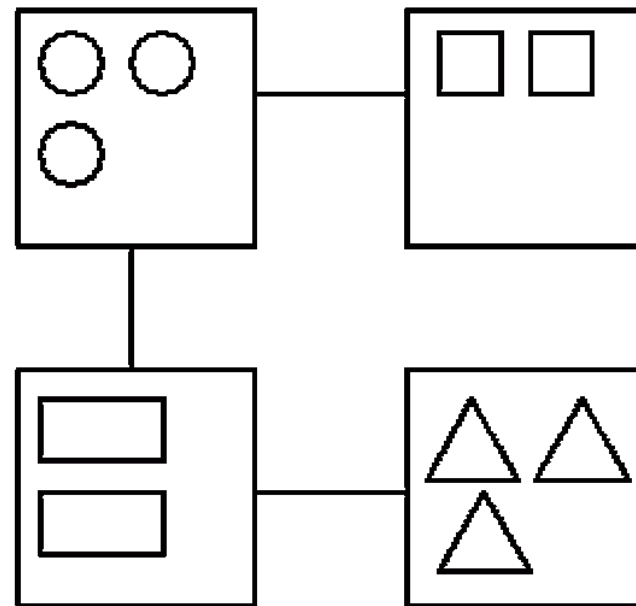
c)... **The best designs** have **high cohesion** (also called strong cohesion) within a module and **low coupling** (also called weak coupling) between modules.

Gợi ý: between modules.

## – Mô tả khái niệm về cohesion và coupling



**Low cohesion and high coupling**

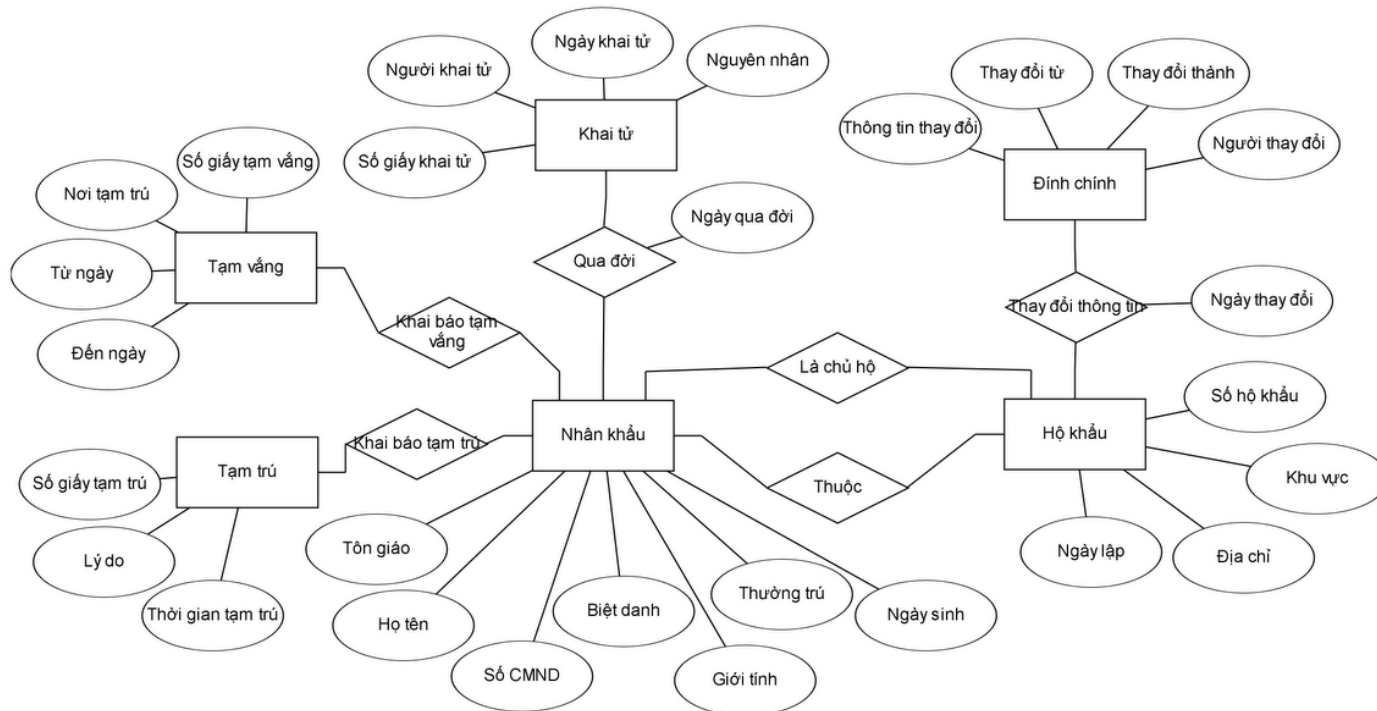


**High cohesion and low coupling**

# Phần II: Thiết kế cho bài toán (case study)

## 1. Thiết kế cơ sở dữ liệu

- Từ mô hình dữ liệu ERD đã xây dựng trong bài thực hành trước, hãy tạo cơ sở dữ liệu và các bảng dữ liệu tương ứng trong hệ quản trị MySQL.
- Nhắc lại: Mô hình dữ liệu ERD cho nhóm chức năng số 1:



# Phần II: Thiết kế cho bài toán (case study)

## Công việc thực hiện:

- Xây dựng các bảng dữ liệu, xác định quan hệ giữa các bảng và thiết kế chi tiết cho mỗi trường (cột) trong bảng.
- Một số lưu ý khi thiết kế các trường dữ liệu: Field là đơn vị nhỏ nhất của dữ liệu → Field tương ứng với 1 thuộc tính (attribute) trong mô hình dữ liệu logic. Quyết định cần làm khi thiết kế là phải **chọn kiểu dữ liệu cho field, kiểm soát tính toàn vẹn dữ liệu** và Hệ quản trị dữ liệu sẽ quản lý các giá trị bị thiếu cho field như thế nào?
- Xây dựng sơ đồ quan hệ giữa các bảng dữ liệu
- Đặc tả thiết kế dữ liệu cho từng bảng

# Phần II: Thiết kế cho bài toán (case study)

## Công việc thực hiện:

Loại đặc tả	Mô tả nội dung
Tên trường (field name)	Theo quy định về cách đặt tên hoặc quy ước viết code của nhóm
Kiểu trường (data type)	Kiểu dữ liệu phù hợp với các giá trị cần lưu trữ trong các trường
Kích cỡ (size)	Kích thước / Khoảng giá trị tối đa và tối thiểu
Mã hóa (Coding)	Mã hoá các thông tin lưu trữ / Khuôn dạng của dữ liệu
Các quy tắc toàn vẹn dữ liệu (data integrity rules)	Đặc tả về các hạn chế đặt lên giá trị của trường: NULL / NOT NULL/ KHOÁ CHÍNH,...
Các kiểm soát bảo trì (maintenance controls)	Chỉ ra những giá trị nào được phép thay đổi
Công thức (Formular)	Mô tả công thức tính toán giá trị với những trường cần tính toán.
Toàn vẹn tham chiếu (references integrity)	Đặc tả giá trị của trường có liên quan đến giá trị của trường khác, ví dụ: KHOÁ NGOẠI,...
Sở hữu (Ownership)	Ai có quyền đối với dữ liệu.

# Phần II: Thiết kế cho bài toán (case study)

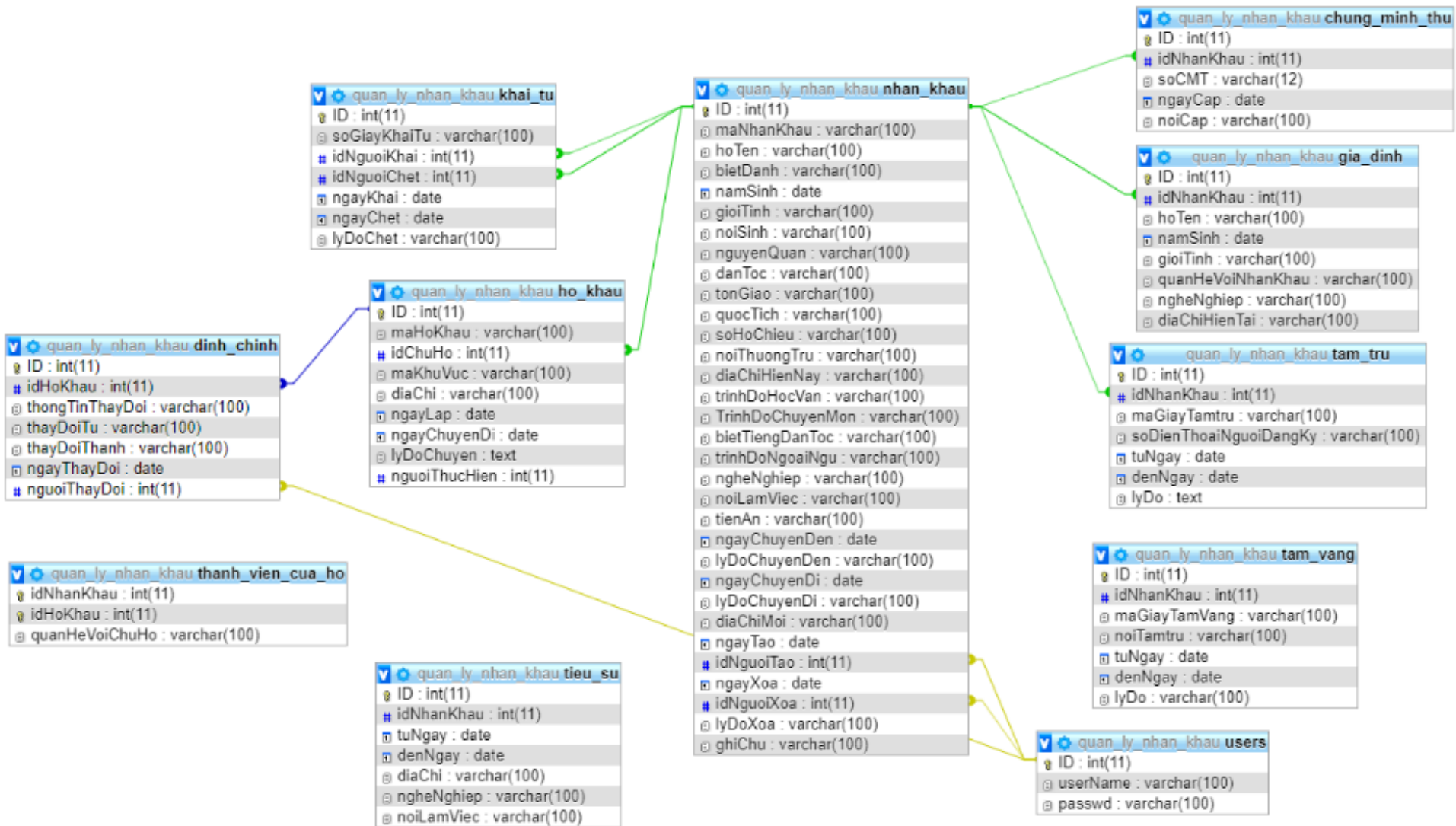
## Mẫu đặc tả thiết kế một bảng dữ liệu:

Tên bảng:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc toàn vẹn	Khuôn dạng	Ghi chú

# Phần II: Thiết kế cho bài toán (case study)

## Ví dụ: Thiết kế cơ sở dữ liệu cho nhóm chức năng số 1:





# Phần II: Thiết kế cho bài toán (case study)

## Ví dụ: Thiết kế cơ sở dữ liệu cho nhóm chức năng số 1:

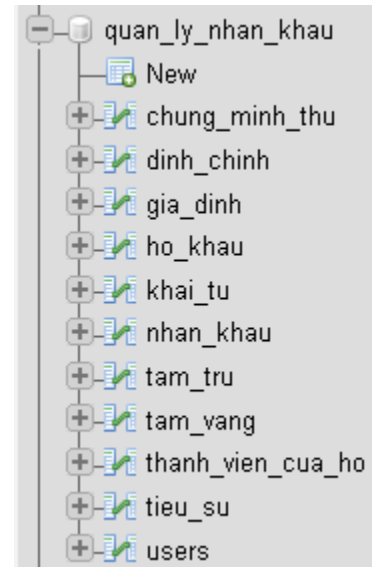
**Bảng HoK Hau:**

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc toàn vẹn	Khuôn dạng	Ghi chú
<u>ID</u>	Int		Khoá chính	Số nguyên dương	
maHoKhu	Varchar(100)	100 ký tự		Văn bản	
idChuHo	Int		Khoá tham chiếu từ bảng NhanKhu	Số nguyên dương	
maKhuVuc	Varchar(100)	100 ký tự		Văn bản	
diaChi	Varchar(100)	100 ký tự		Văn bản	
ngayLap	Date			Ngày tháng năm	
ngayChuyenDi	Date			Ngày tháng năm	
lyDoChuyen	Text	Văn bản dài		Văn bản	
nguoiThucHien	Int		Mã tài khoản thêm thông tin	Số nguyên dương	

# Phần II: Thiết kế cho bài toán (case study)

## Import cơ sở dữ liệu mẫu vào hệ quản trị cơ sở dữ liệu:

- Bước 1: Mở công cụ phpMyAdmin
- Bước 2: Tạo một CSDL mới có tên là **quan\_ly\_nhan\_khau**, chọn CSDL này để import các bảng và dữ liệu mẫu
- Bước 3: Chọn chức năng Import trên thanh công cụ
- Bước 4: Nhấn vào nút Choose File và chọn tệp **quan\_ly\_nhan\_khau.sql** (giảng viên cung cấp)
- Bước 5: Hoàn thành việc import cơ sở dữ liệu



# Phần II: Thiết kế cho bài toán (case study)

**Bài tập:** Xây dựng cơ sở dữ liệu và đặc tả thiết kế các bảng dữ liệu cho bài toán với nhóm chức năng đã chọn.

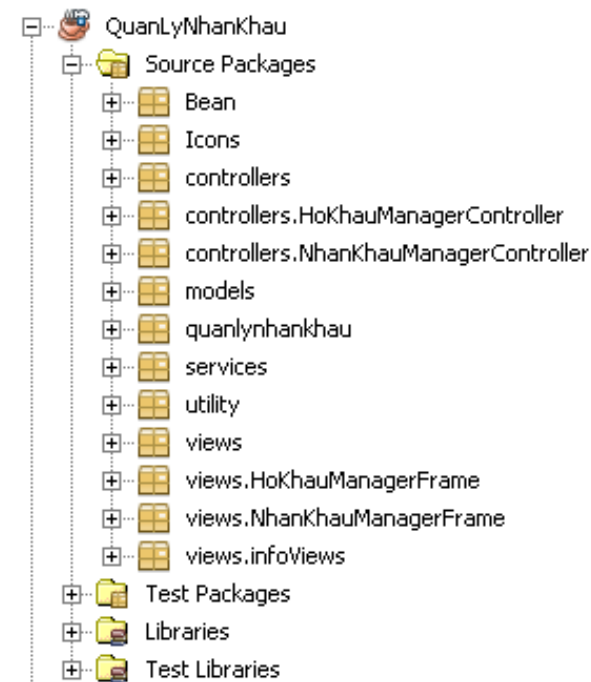
**Các nhóm có thể sử dụng Cơ sở dữ liệu có sẵn đã cung cấp và thiết kế thêm các bảng cần thiết cho nhóm chức năng của phần mềm hoặc thiết kế một Database mới.**

**Trong cả hai cách tiếp cận đều phải đặc tả thiết kế chi tiết cho dữ liệu như đã hướng dẫn ở phần trên.**

# Phần II: Thiết kế cho bài toán (case study)

## 2. Kiến trúc của phần mềm và thiết kế chi tiết các lớp

- Import Project mẫu
  - Bước 1: Chọn tệp **QuanLyNhanKhau.zip** (giảng viên cung cấp) giải nén
  - Bước 2: Mở công cụ NetBeans IDE
  - Bước 3: Chọn menu File → Open Project và mở project trong thư mục đã giải nén ở trên



# Phần II: Thiết kế cho bài toán (case study)

## 2. Kiến trúc của phần mềm và thiết kế chi tiết các lớp

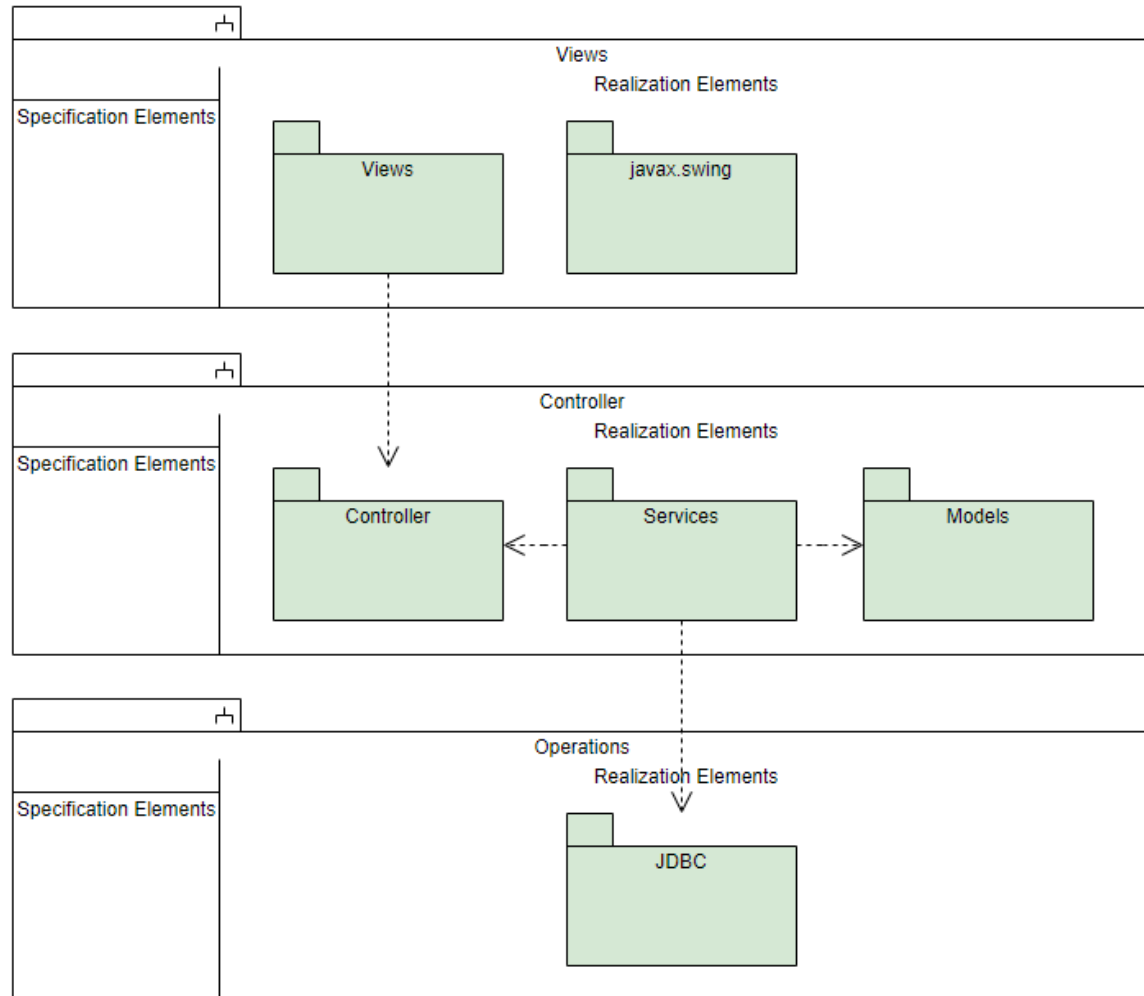
- Project chia thành các gói (package) như sau:
  - **package views**: chứa các lớp kế thừa từ javax.swing.\* định nghĩa các màn hình giao diện người dùng cho ứng dụng
  - **package controllers**: chứa các lớp điều khiển, xử lý nghiệp vụ chính
  - **package models**: định nghĩa các cấu trúc dữ liệu sử dụng trong phần mềm
  - **package services**: chứa các lớp tác động đến dữ liệu trong cơ sở dữ liệu (cung cấp các phương thức thực hiện truy vấn SELECT, INSERT, UPDATE, DELETE trên các bảng dữ liệu)
  - **package Bean**: chứa các lớp Bean đóng gói các thực thể dữ liệu
  - **package utility**: chứa các lớp tiện ích

# Phần II: Thiết kế cho bài toán (case study)

**Bài tập:** Xây dựng sơ đồ gói (package diagram) mô tả kiến trúc logic của ứng dụng.

## Gợi ý:

Xây dựng sơ đồ gói có dạng như sau, đặt các lớp theo tổ chức của Project vào các package phù hợp trên sơ đồ



# Phần II: Thiết kế cho bài toán (case study)

**Bài tập: Xây dựng (viết code và đặc tả thiết kế) các lớp trong package model tương ứng với các thực thể dữ liệu mà nhóm đã thiết kế trong Database cho nhóm chức năng đã chọn trong bài toán.**

# Phần II: Thiết kế cho bài toán (case study)

## Gợi ý:

Các class dạng này sẽ bao gồm khai báo các thuộc tính và các phương thức get / set trên các thuộc tính này (đảm bảo đóng gói và che giấu thông tin)  
Class **HoKhouModel** như sau:

```
package models;
import java.util.Date;
public class HoKhouModel {
    private int ID;
    private String maHoKhou;
    private int idChuHo;
    private String maKhuVuc;
    private String diaChi;
    private Date ngayLap;
    private Date ngayChuyDi;
    private String lyDoChuyen;
    private int nguoiThucHien;
    public int getID() { ...3 lines }
    public void setID(int ID) { ...3 lines }
    public String getMaHoKhou() { ...3 lines }
    public void setMaHoKhou(String maHoKhou) { ...3 lines }
    public int getIdChuHo() { ...3 lines }
    public void setIdChuHo(int idChuHo) { ...3 lines }
    public String getMaKhuVuc() { ...3 lines }
    public void setMaKhuVuc(String maKhuVuc) { ...3 lines }
    public String getDiaChi() { ...3 lines }
    public void setDiaChi(String diaChi) { ...3 lines }
    public Date getNgayLap() { ...3 lines }
    public void setNgayLap(Date ngayLap) { ...3 lines }
    public Date getNgayChuyDi() { ...3 lines }
    public void setNgayChuyDi(Date ngayChuyDi) { ...3 lines }
    public String getLyDoChuyen() { ...3 lines }
    public void setLyDoChuyen(String lyDoChuyen) { ...3 lines }
    public int getNguoiThucHien() { ...3 lines }
    public void setNguoiThucHien(int nguoiThucHien) { ...3 lines }
}
```




























# Phần II: Thiết kế cho bài toán (case study)

## Gợi ý:

Xây dựng mô tả chi tiết cho class theo mẫu: “CRC Card Layout”

Class Name	
Responsibilities	Collaborators

Class <b>HoKhouModel</b>	
Chứa các thông tin về sổ hộ khẩu: private int ID; private String maHoKhou; private int idChuHo; private String maKhuVuc; private String diaChi; private Date ngayLap; private Date ngayChuyDi; private String lyDoChuyen; private int nguoiThucHien;	Class <b>HoKhouController</b>

 <b>HoKhouModel</b>	
 - int ID  - String maHoKhou  - int idChuHo  - String maKhuVuc  - String diaChi  - Date ngayLap  - Date ngayChuyDi  - String lyDoChuyen  - int nguoiThucHien	 + int getID()  + void setID(int ID)  + String getMaHoKhou()  + void setMaHoKhou(String maHoKhou)  + int getIdChuHo()  + void setIdChuHo(int idChuHo)  + String getMaKhuVuc()  + void setMaKhuVuc(String maKhuVuc)  + String getDiaChi()  + void setDiaChi(String diaChi)  + Date getNgayLap()  + void setNgayLap(Date ngayLap)  + Date getNgayChuyDi()  + void setNgayChuyDi(Date ngayChuyDi)  + String getLyDoChuyen()

# Phần II: Thiết kế cho bài toán (case study)

- **Yêu cầu:**

- Hoàn thành Thiết kế cho bài toán (casestudy) với các nội dung: thiết kế và đặc tả cơ sở dữ liệu, sơ đồ package mô tả kiến trúc logic của ứng dụng, thiết kế các lớp model tương ứng với các bảng trong CSDL đã xây dựng.
- Phần nội dung này các nhóm làm vào trong file .docx (báo cáo)
- Các nhóm chuẩn bị thêm một **slide powerpoint** về nội dung Thiết kế ở trên, buổi học tiếp theo sẽ trình bày.